common
EUROPE LUXEMBOURG
Modernisation des Applications IBM i & Open Source
(webinaire)
Jeudi 25 mars – 14h00 – 17h00

Ansible Automation on IBM i

Benoit MAROLLEAU – Cloud/AI Architect
IBM Systems Center Europe, Montpellier, France
benoit.marolleau@fr.ibm.com

Replays & Presentations
https://ibm.biz/bma-wiki

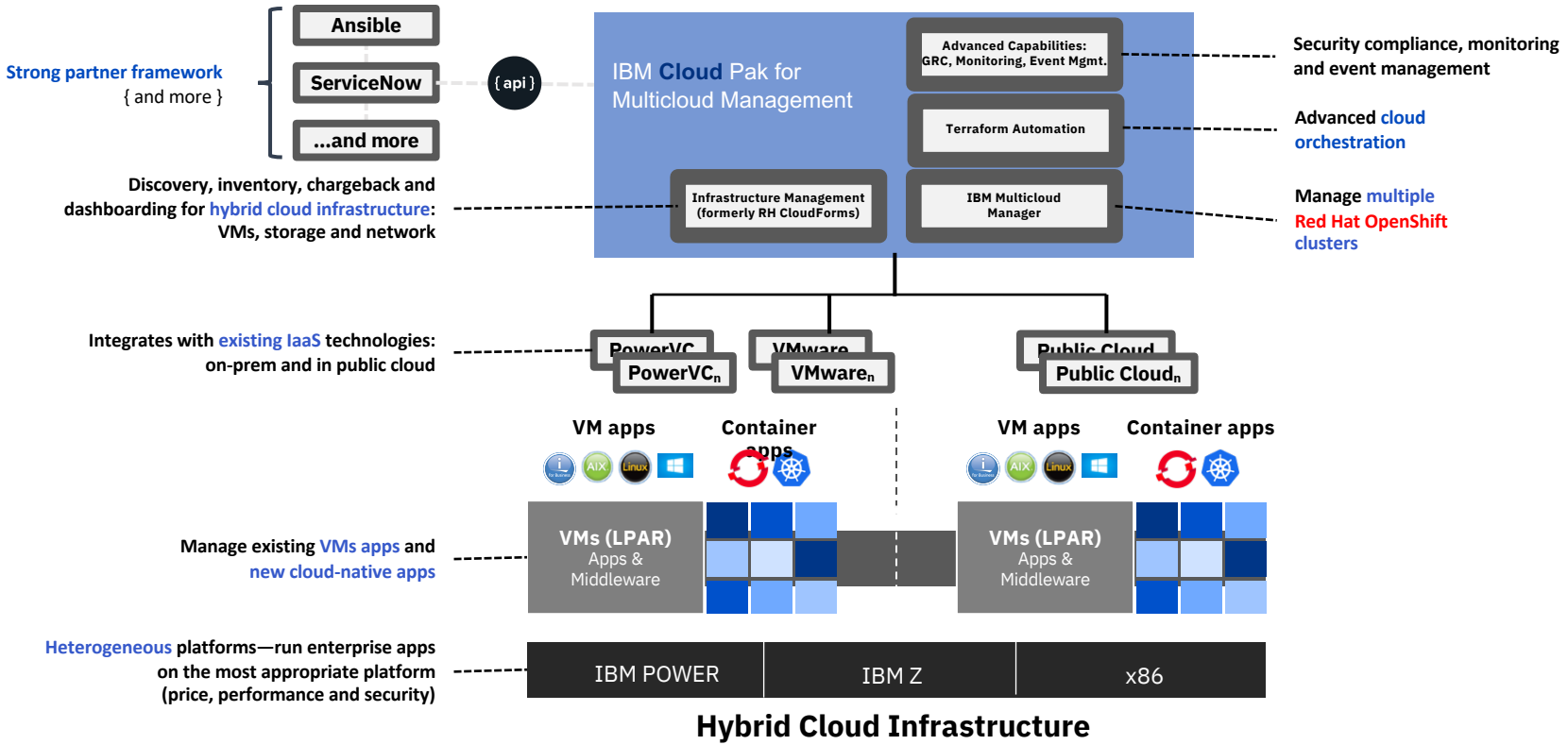
Agenda

- Infrastructure as Code Strategy
- Ansible 101
- Ansible & IBM i
- Demonstration

Infrastructure as Code ?

https://www.ibm.com/cloud/blog/chef-ansible-puppet-terraform

Hybrid Cloud Architecture with IBM **Cloud** Pak for Multicloud Management

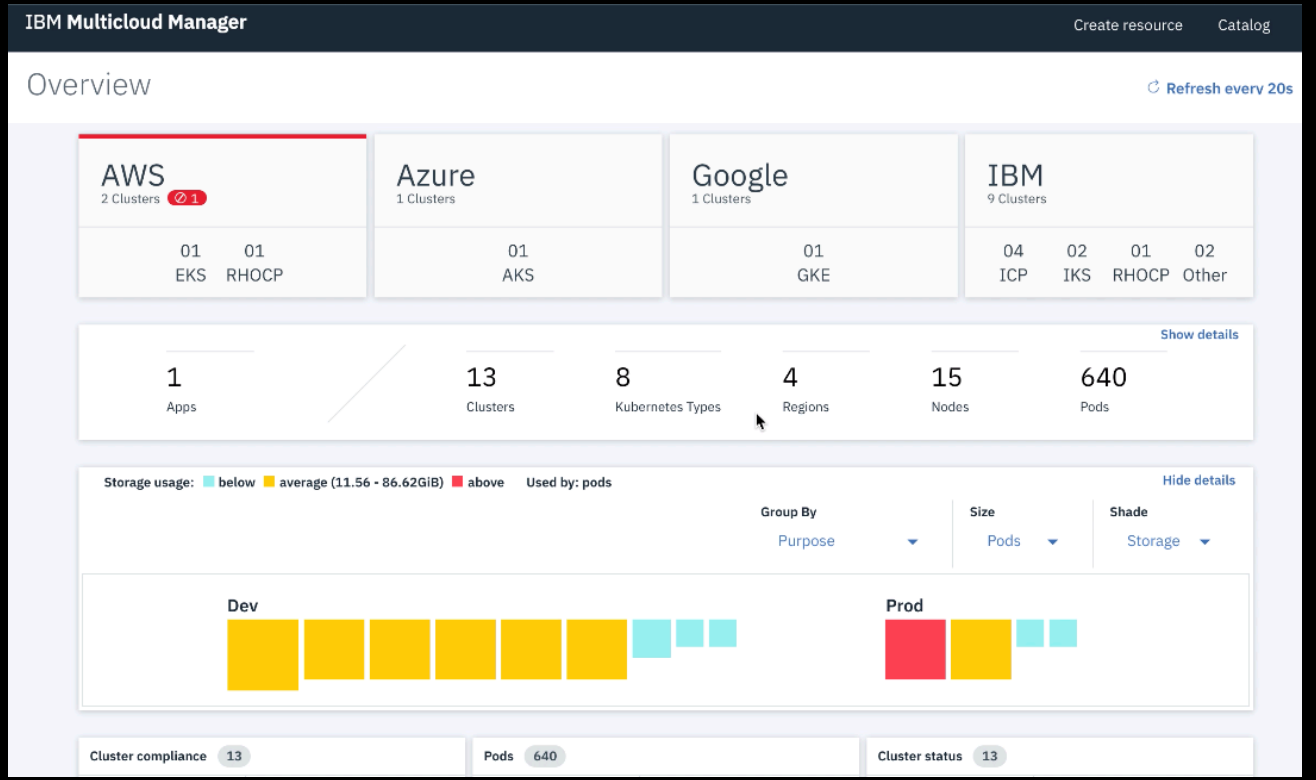


IBM Multicloud Manager

Visibility
- Development teams can see all the deployed pods
- IT Operations can see clusters and nodes

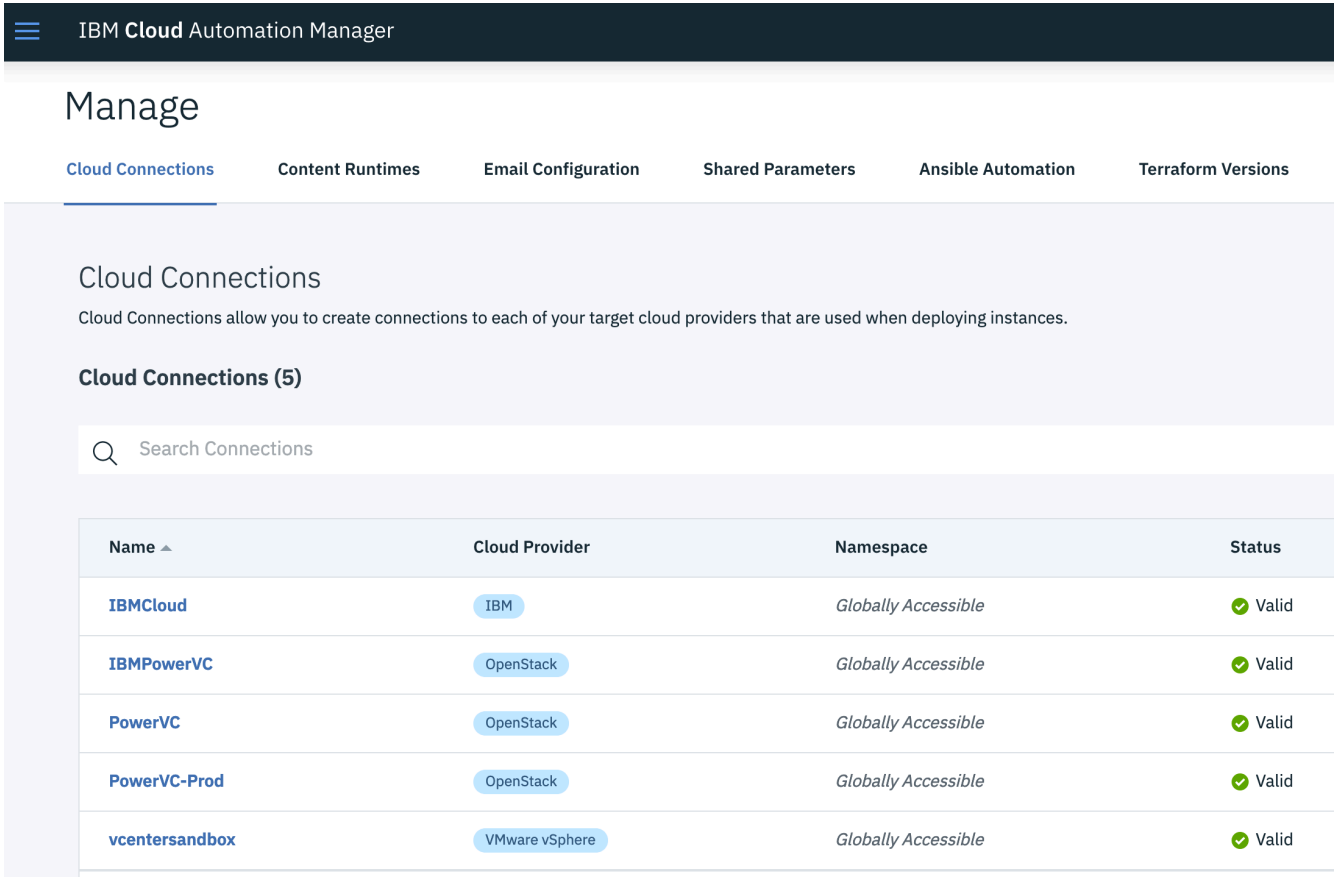
Security and Governance
- Consistent configuration and security policies across cluster

Automation
- Automatically provision, configure and deliver additional Kubernetes clusters in any cloud environment supported by Cloud Automation Manager

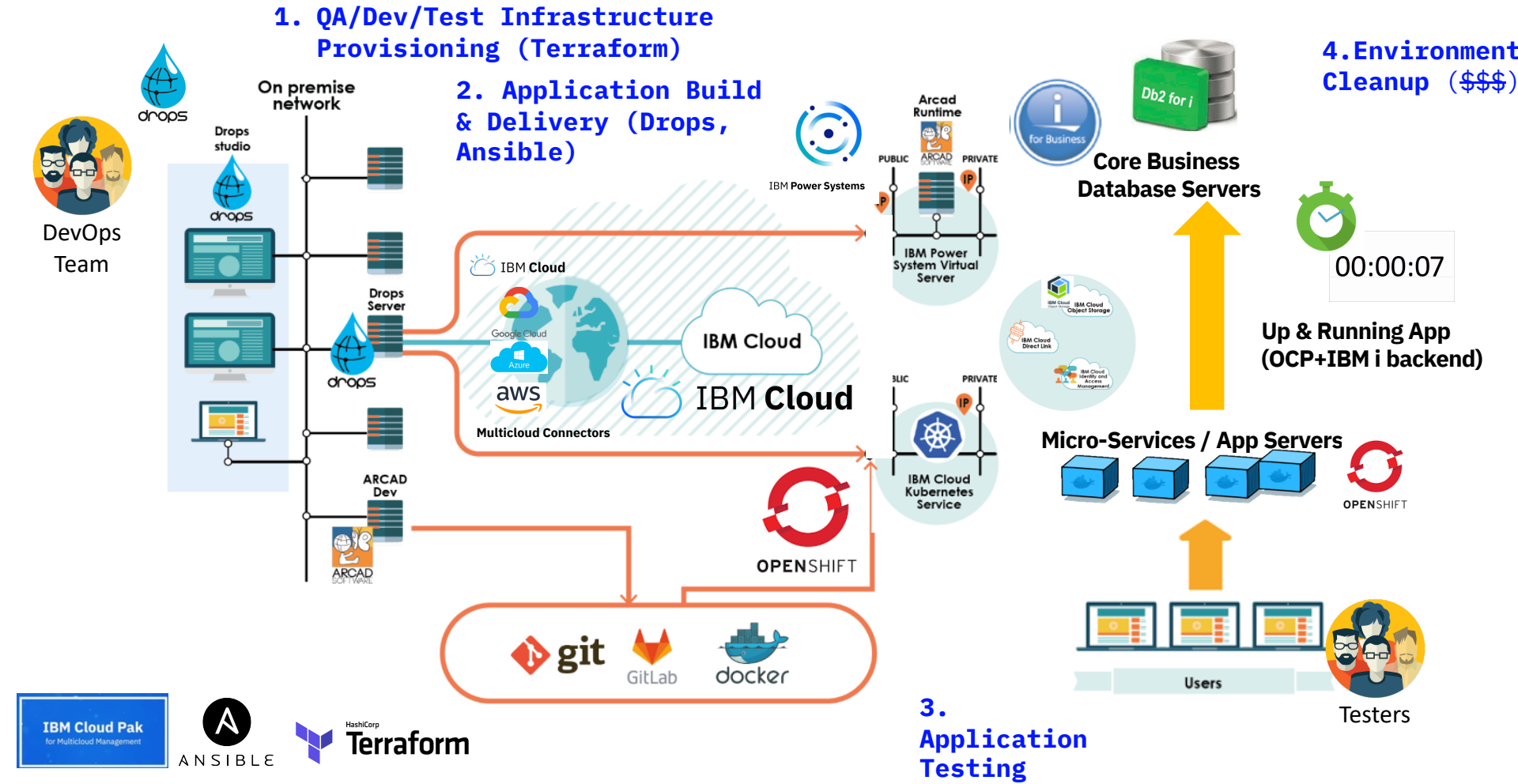


Cloud Pak for MCM

Multiple Cloud Connectors : VMWare, PowerVC, Public Clouds



MultiCloud CI/CD with Arcad Drops on IBM Cloud / Power Virtual Server



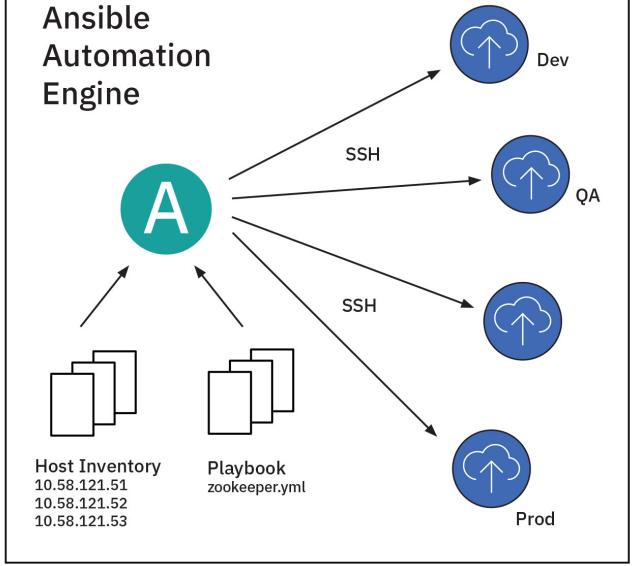
Ansible

Introduction to Ansible

Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy.

- Free open source application
- **Agent-less** – No need for agent installation and management
- Python/YAML based
- Highly flexible and configuration management of systems.
- Configuration roll-back in case of error

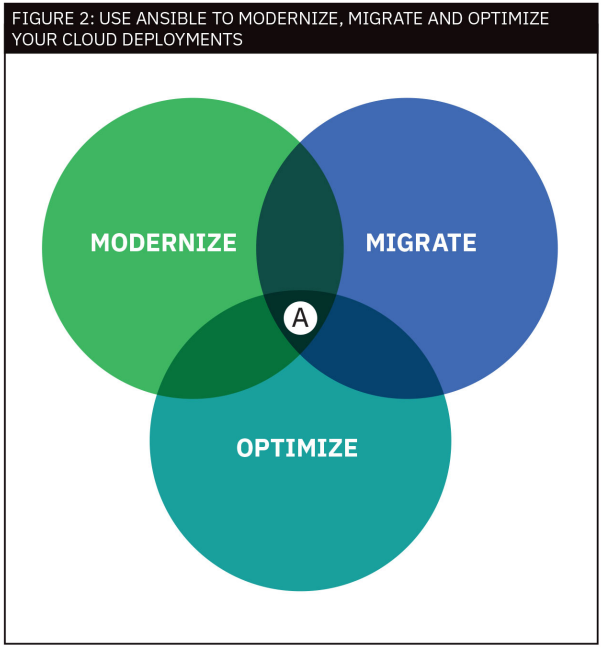
FIGURE 1: HOW ANSIBLE WORKS



Introduction to Ansible

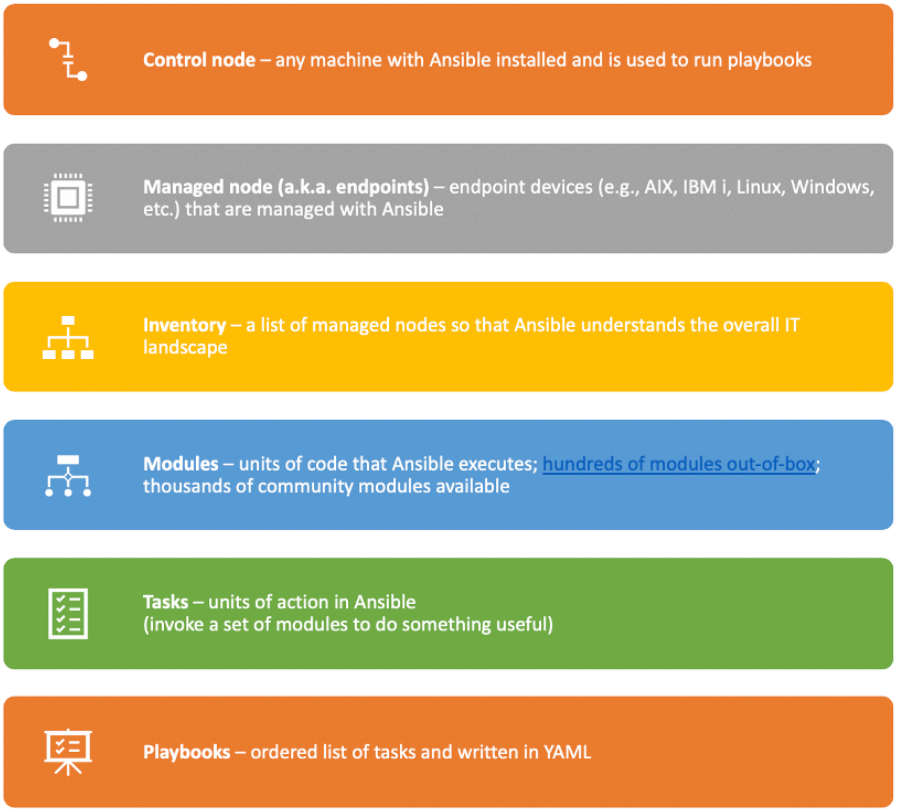
Common use cases are:

- ✓ Automation of repeatable operations tasks like **system administration**, code build, deployment and QA tests in your DevOps cloud infrastructure and cloud deployment pipeline
- ✓ **Cloud infrastructure and application provisioning**, complementing other infrastructure as code technologies like Terraform
- ✓ **Configuration and security compliance** reinforcement by checking and fixing systems settings versus policies in place



IBM i Operations like any other platforms…
Efficient for new hires onboarding !!

Introduction to Ansible



Introduction to Ansible

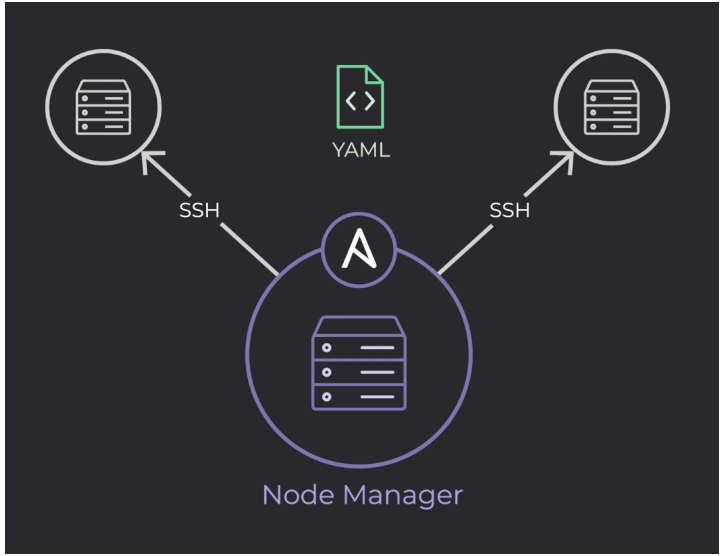
Idempotency , Parallelism

Ansible is **idempotent**, it means that an operation has the same effect, whether applied one or more times. If the state requested in the action is in accordance with what is requested, then Ansible does nothing.

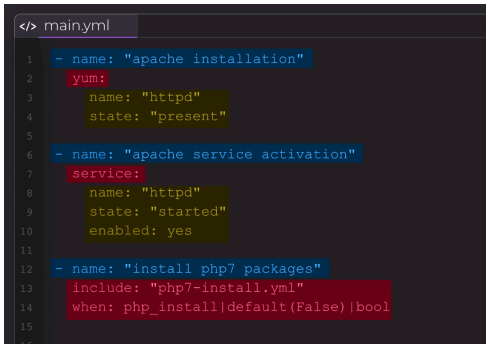
Idempotency is possible because Ansible manages 3 execution states for a task : **skipping**, **ok** or **changed**.

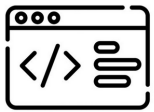
Ansible launches the tasks in push mode (in parallel) on all the servers at the same time. On the other hand, the tasks are executed sequentially one after the other.

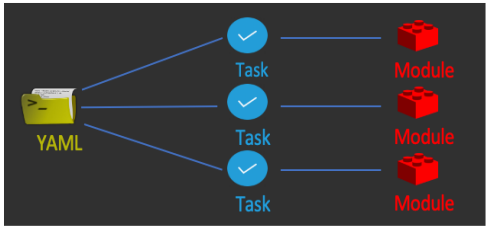
The first task will be sent and executed on all servers before proceeding to the next task. This means that the node manager waits for a response from all the nodes and waits until they have all completed the execution of the current task before proceeding to the next one.



Introduction to Ansible







YAML (*Yet Another Markup Language*)

→ Each task invokes a module (python program). You can have multiple modules per task.

What is Red Hat Ansible Engine?

Ansible Engine provides the core, agentless functionality of Ansible that everything else builds upon

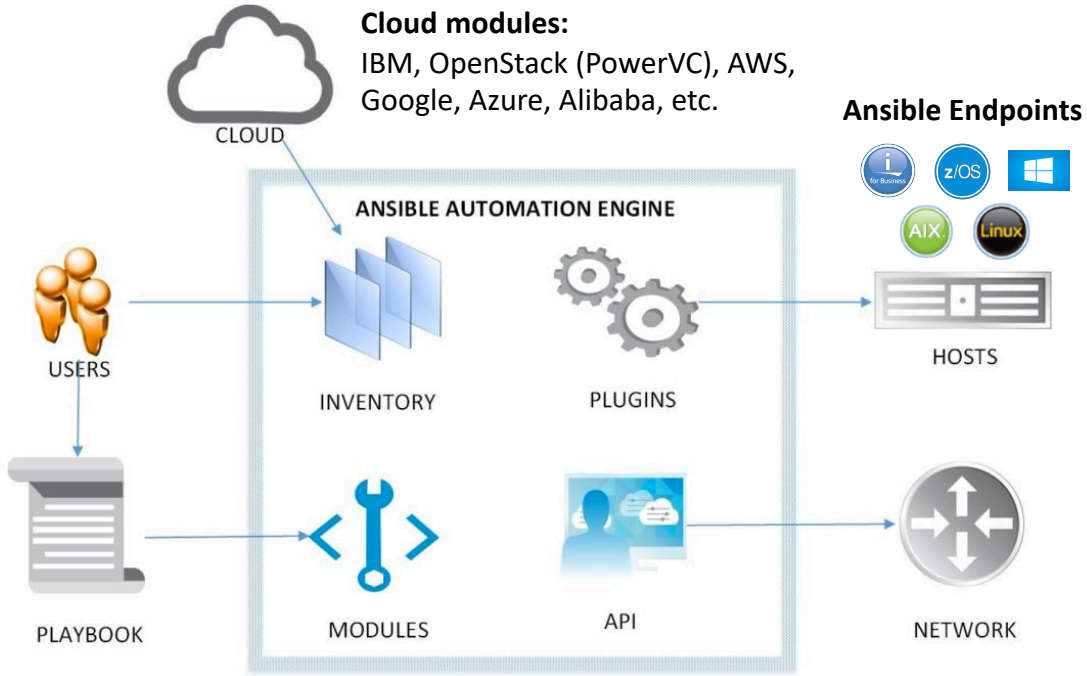
Includes the basic building blocks of Ansible—the control node, managed nodes (endpoints), inventory, modules, tasks and playbooks

Commercial form of Ansible technology

Available for subscription purchase from Red Hat—from a POWER perspective, includes enterprise support options for AIX and IBM i managed endpoints (in June 2020)

Red Hat Ansible CERTIFIED INTEGRATION: **Ansible and IBM Power Systems**

https://www.ansible.com/integrations/infrastructure/ibm-power-systems



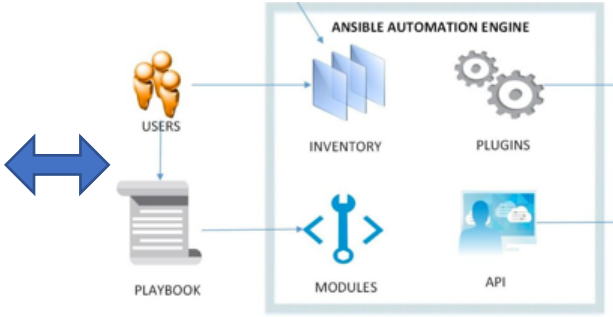
RED HAT® ANSIBLE® Engine

Red Hat Ansible Engine supported on x86 Linux only — manages to endpoints

Ansible Automation for IBM Power Systems

RED HAT ANSIBLE Tower

RED HAT ANSIBLE Engine



Endpoints

1 Red Hat Ansible Tower

- Enterprise-wide graphical control of Ansible estate

Supported on Linux

2 Red Hat Ansible Engine

- Enterprise-wide control – i.e., runs playbooks

Supported on Linux

3 Red Hat Ansible Endpoints

- Enterprise-wide automation; modules are executed here

Commercial support available from Red Hat

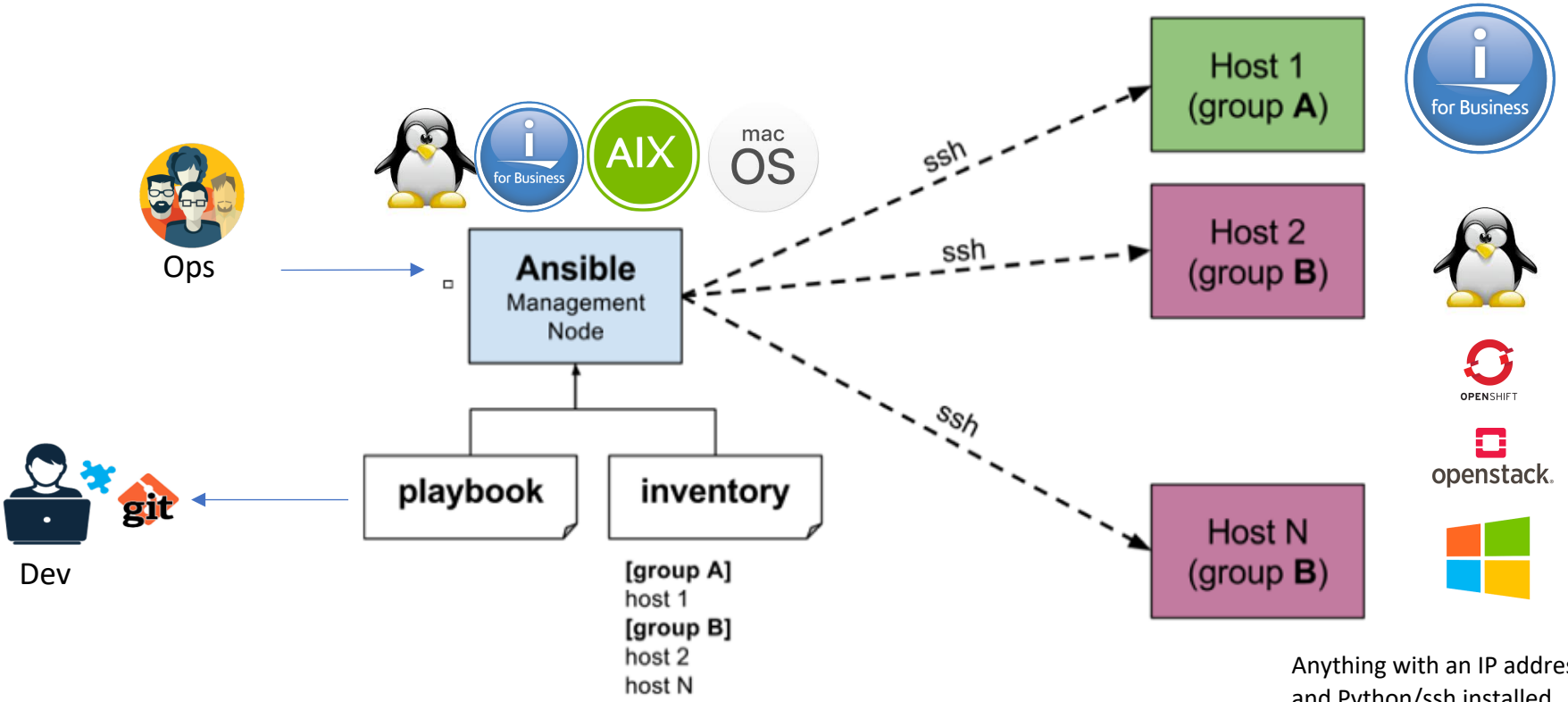
community support (at present)

Ansible & IBM i

100 000 ‘s Customers
120 countries
100,000’s Systems
70% Small and Mid-sized
40 national languages
98% of Fortune 100

Ansible Solution Architecture

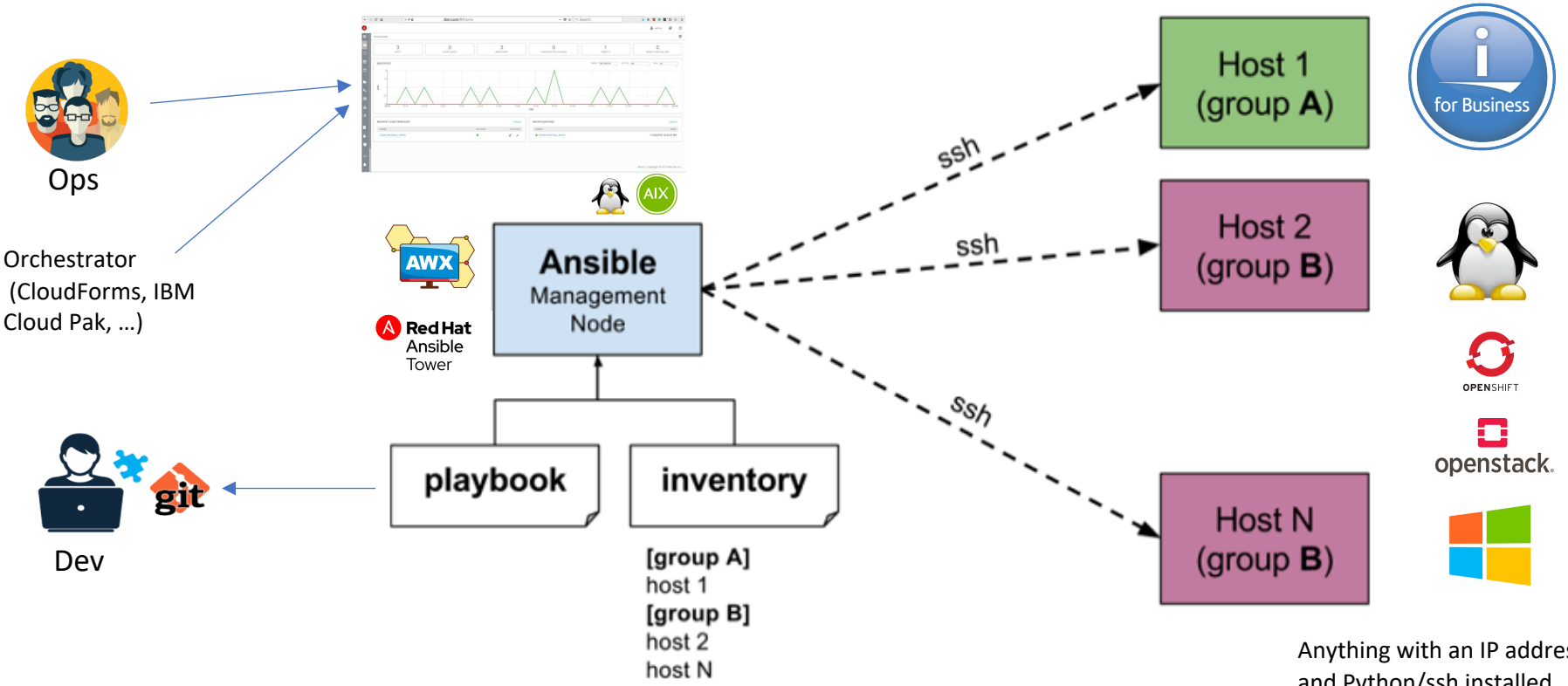
Basic Setup



Anything with an IP address and Python/ssh installed

Ansible Solution Architecture

Advanced Setup



Ansible Tower

Red Hat Ansible Tower

for Business

Ansible Tower = Supported Ansible solution for Enterprise

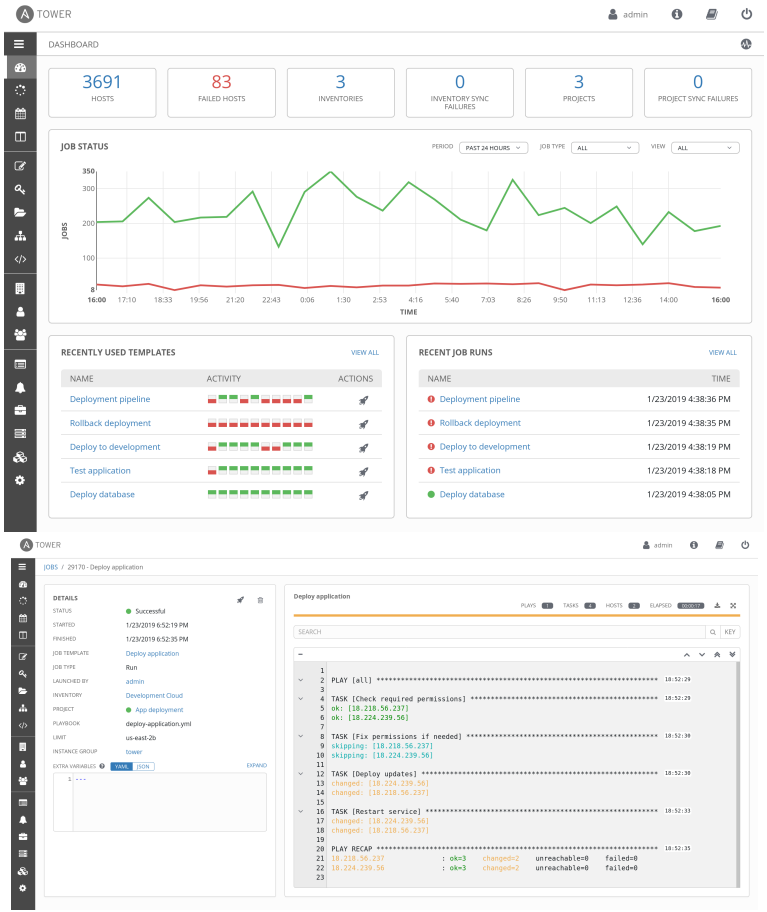
YOUR ANSIBLE DASHBOARD

REAL-TIME JOB STATUS UPDATES

MULTI-PLAYBOOK WORKFLOWS

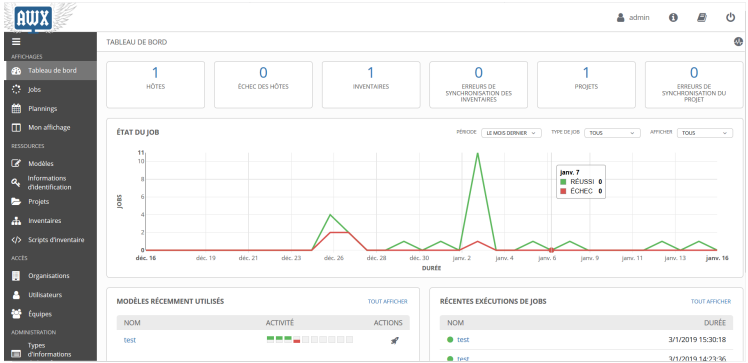
WHO RAN WHAT JOB WHEN

SUPPORT IBM i TASKS

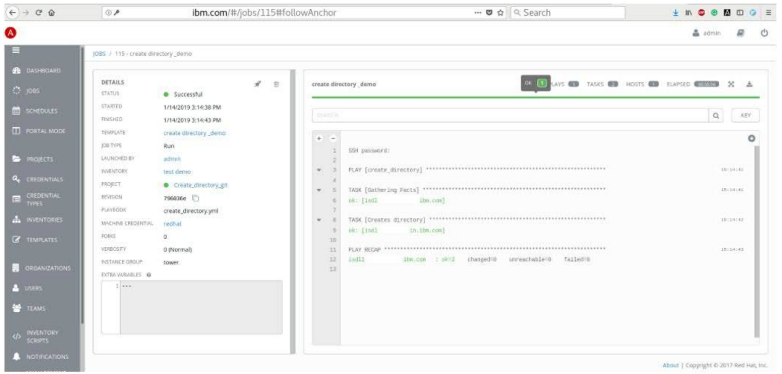


Ansible AWX

https://github.com/ansible/awx



Ansible Tower upstream project. Community Support



YOUR ANSIBLE DASHBOARD

REAL-TIME JOB STATUS UPDATES

MULTI-PLAYBOOK WORKFLOWS

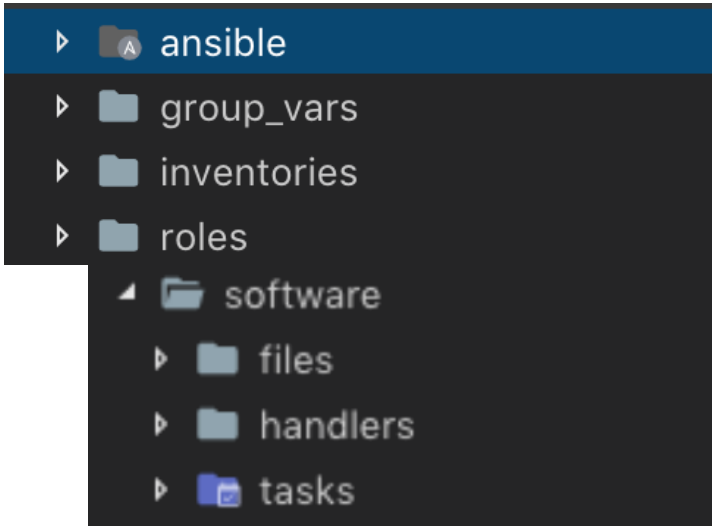
WHO RAN WHAT JOB WHEN

SUPPORT IBM i TASKS

https://developer.ibm.com/technologies/systems/articles/automation-using-ansible-awx-gui/

https://www.power-devops.com/post/installing-ansible-awx-on-ibm-aix

How to use Ansible - Intermediate



group_vars: is the folder which will contain variable with it's value. Which you can use in your script. Variable can use username, password, software version, path or anything.

inventories: Inventories are definition for your cluster, VMs or Nodes. You can define you IP address or Domain name for all your VMs. In-case you want to SSH using public key or password based Authentication. Default : /etc/ansible/hosts

roles: This is the place where you define your automation.
Example:
IBMi > **files** (contains SAVF)
IBMi > **tasks** (contains your Playbooks i.e. your tasks in a script)
IBMi > **handlers** running operations on change: sometimes, you want a task to run only when a change is made on a machine Handlers are tasks that only run when notified.

ansible.cfg : system wide or user ansible config file
Usually in /etc/ansible/ansible.config or ~/.ansible.cfg

Ansible and IBM i : Modules

Write your first “playbook” in YAML format to describe what you want on your managed node inventory and Ansible will , for example :

- ✓ Deploy or clone a new environment on an IBM i VM on either a private or public cloud
- ✓ Install a new licensed program product or application version containing libraries, database and IFS artifacts
- ✓ Save or restore objects, manage servers or jobs and check and install PTFs
- ✓ Control your security settings, like managing user profiles and authorities, or check IFS rights. Ansible gathers facts and can remediate any security deviations.
- ✓ Orchestrate all of the above or a subset of these tasks

Ansible and IBM i

Core modules in PASE + IBM i Specific Modules

Core Maintained modules are maintained by the Ansible Engineering Team.

- Core modules are owned by RedHat and ship with Ansible installation.
- Many of these modules work for IBM i PASE environment.
- Support PASE but not native IBM i.

- command
- raw
- script
- shell
- pip
- yum
- pause
- wait_for_connection
- ~~at~~
- authorized_key
- gather_facts
- ~~group~~
- ~~Mount~~
- ping
- ~~reboot~~
- setup
- ~~user~~
- assemble
- blockinfile
- copy
- fetch
- file
- find
- lineinfile
- stat
- synchronize
- git

Ansible and IBM i

Core modules in PASE + **IBM i Specific Modules**

- CL Commands
  - Executes CL commands and return general and detail job logs
- SQLs executions
  - Executes SQL statements and return the results
  - Queries – compare the returned single value result
  - Inserts / Updates / Deletes
  - Functions & Procedures
- Gathering facts and setup changes for IBM i
- Securities – authorization list, user profiles, grant object authorities
- Copy Objects, Fetch Objects, Find Objects
- Reply Message – query and reply
- Reboot system
- Network configurations
- Device configurations and management
- IASP configuration
- System Values, Environment variables, Etc.
- Submit / Schedule Jobs
- Manage fixes / PTFs / LPPs

- …. More to come!!! Check out

https://github.com/IBM/ansible-for-i

ibmi_at
Schedule a batch job on a remote IBMi node.
ibmi_cl_command
Executes a CL command.
ibmi_copy
Copy a save file from local to a remote IBMi node.
ibmi_display_subsystem
Display all currently active subsystems or currently active jobs in a subsystem.
ibmi_end_subsystem
End a subsystem.
ibmi_fetch
Fetch objects or a library from a remote IBMi node and store on local.
ibmi_install_product_from_savf
Install the the licensed program(product) from a save file.
ibmi_lib_restore
Restore one library on a remote IBMi node.
ibmi_lib_save
Save one libary on a remote IBMi node.
ibmi_object_authority
Grant, Revoke and Display the Object Authority.
ibmi_object_restore
Restore one or more objects on a remote IBMi node.

Ibmi_object_save
Save one or more objects on a remote IBMi node.
ibmi_reboot
Reboot IBMi machine.
ibmi_save_product_to_savf
Save the the licensed program(product) to a save file.
ibmi_script
Execute a local cl/sql script file on a remote ibm i node.
ibmi_script_execute
Execute a cl/sql script file on a remote ibm i node.
ibmi_sql_execute
Executes a SQL non-DQL(Data Query Language) statement.
ibmi_sql_query
Executes a SQL DQL(Data Query Language) statement.
ibmi_start_subsystem
Start a subsystem.
ibmi_sync
Synchronize a save file from current ibm i node A to another ibm i node B.
ibmi_synchronize
Synchronize a save file from ibm i node A to another ibm i node B.
ibmi_uninstall_product
Delete the objects that make up the licensed program(product).
ibmi_user_and_group
Create, Change and Display a user(or group) profile.

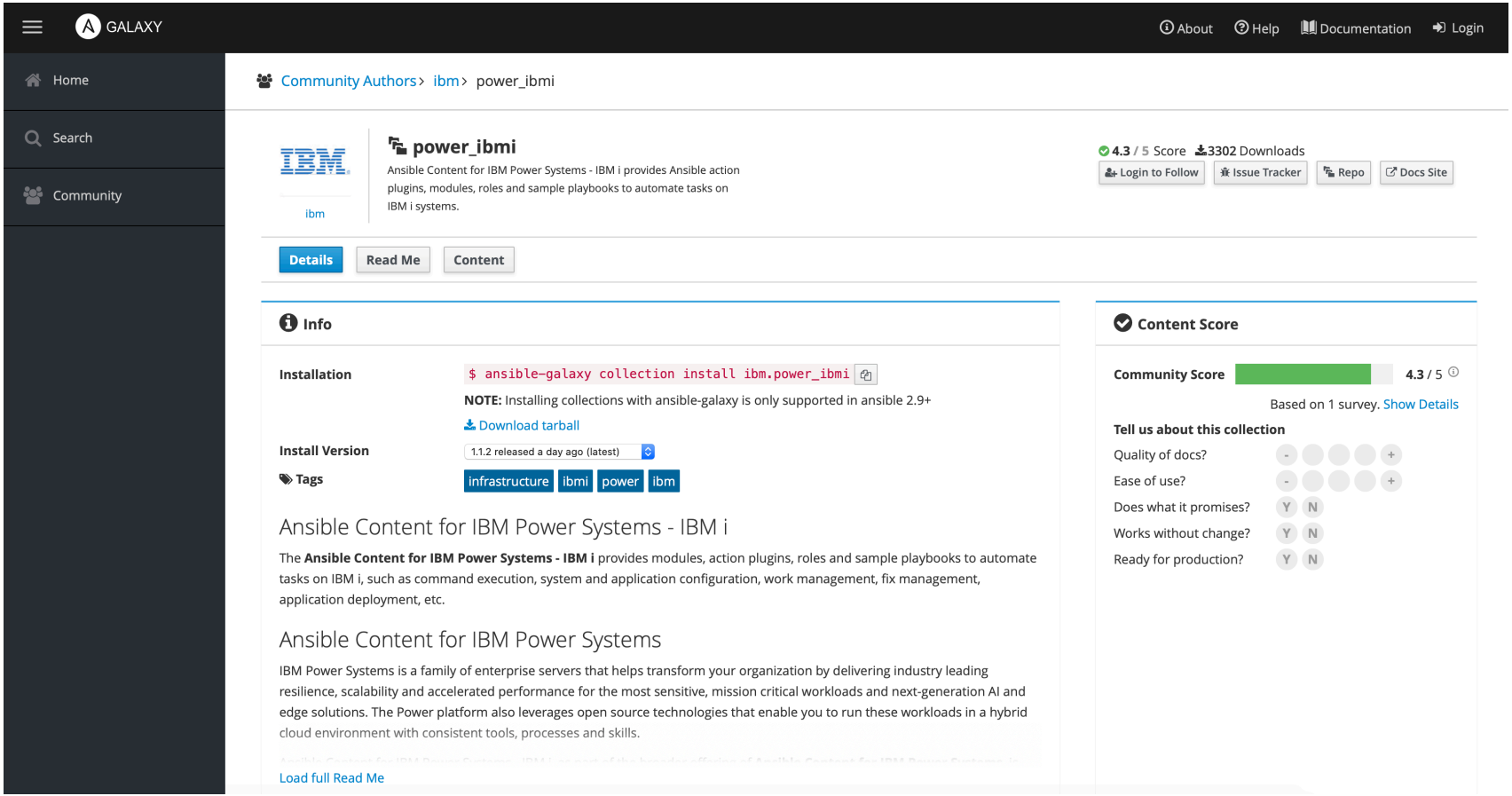
Ansible and IBM i

Playbooks Examples

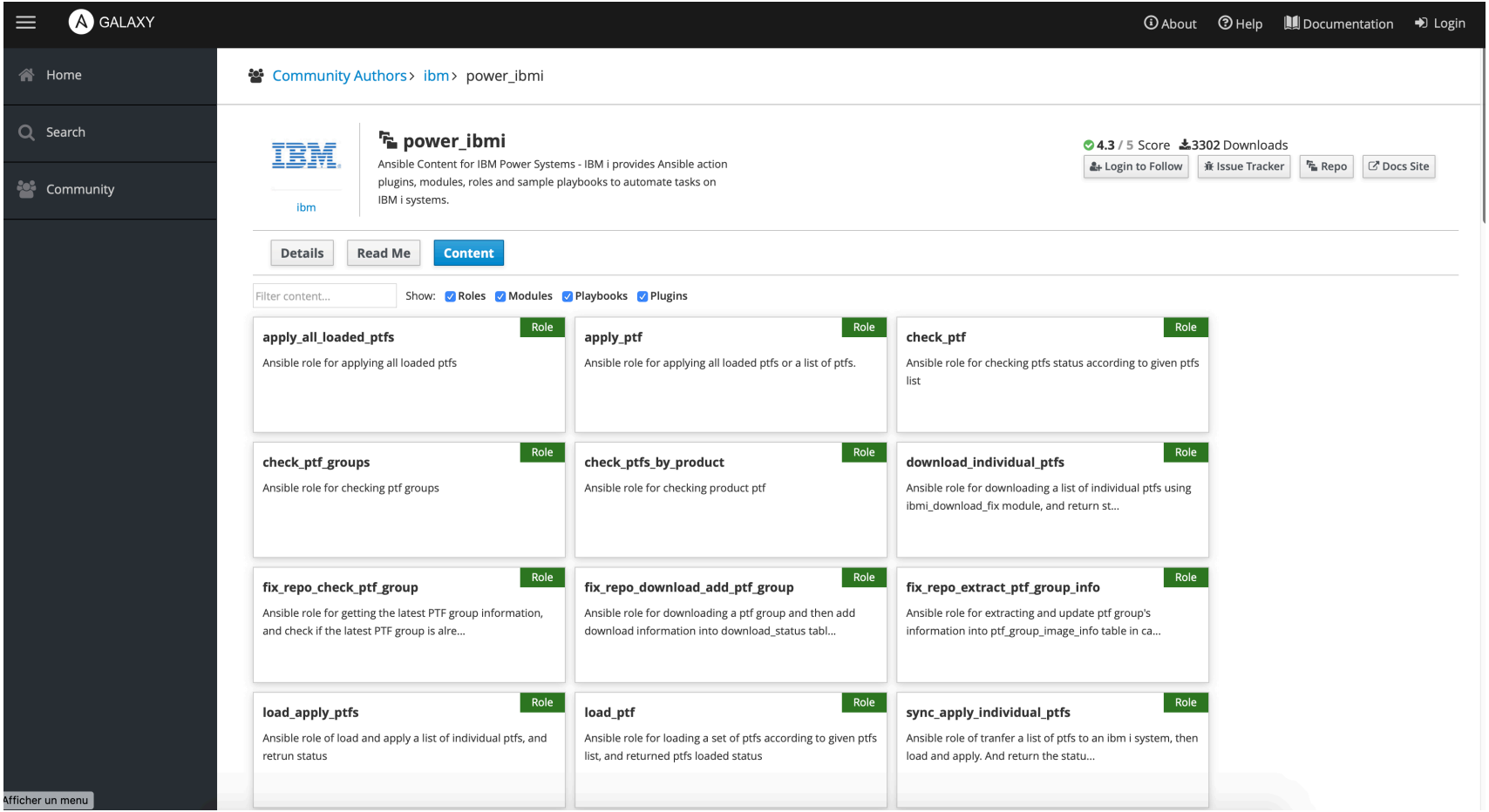
- enable-ansible-for-i
  - ibmi-install-rpm.yml
  - ibm-install-yum.yml
  - setup.yml
- ibmi-install-nodejs
  - ibmi-install-nodejs.yml
- ibmi-check-default-passwords.yml
- ibmi-cl-command-sample.yml
- ibmi-fix-group-check.yml
- ibmi-fix-repo-cum-package.yml
- ibmi-sysval-sample.yml
- query-iasp-sample.yml
- ibmi-sql-sample.yml

https://github.com/IBM/ansible-for-i

Galaxy – power_ibmi



Galaxy – power_ibmi



Ansible Support & Installation

- Ansible on Linux x86 : Community + Possible Red Hat Subscription and support
- Ansible on IBM i : Community + Possible IBM TSS Support (Open Source package)
- Ansible can be installed via your Linux distribution package manager
  - yum install ansible or apt install ansible
  - If not available, just install python-pip and dependencies and install it with “pip”
    - pip install ansible
- Clone the repository to your Ansible server
  - https://github.com/IBM/ansible-for-i
- Create your inventory file
  - example can be found in file examples/ibmi/host_ibmi.ini

Demo

https://github.com/bmarolleau/ansible-for-i

Thank you

Demonstration : Terraform with PowerVC / IBM Cloud, Self service with Cloud Automation Manager / Cloud Pak for MCM

IBM Power Systems in the Multicloud Era : AIX/IBM i Terraform Automation and IBM Cloud Pak for MCM
Part 2 : Automation with Terraform, IBM Cloud Pak for MCM (40 minutes)
https://youtu.be/fHZjQiXSICQ

YouTube

GitHub

https://github.com/bmarolleau/devops-automation/tree/master/terraform/simple-vm-power-vs
https://github.com/bmarolleau/pvs-terraform-lab

Assets & Videos:

- **IBM Power Systems in the Multicloud Era : Get Started with IBM Cloud Power Virtual Server**

Part 1 : Introduction demo to the IBM Cloud Power Virtual Server for AIX / IBM i (20 minutes)

https://youtu.be/RywSfXT_LLs

- **Ansible & Power Systems : IBM i demo**

MOP Demo – Ansible + IBM i

Thank you